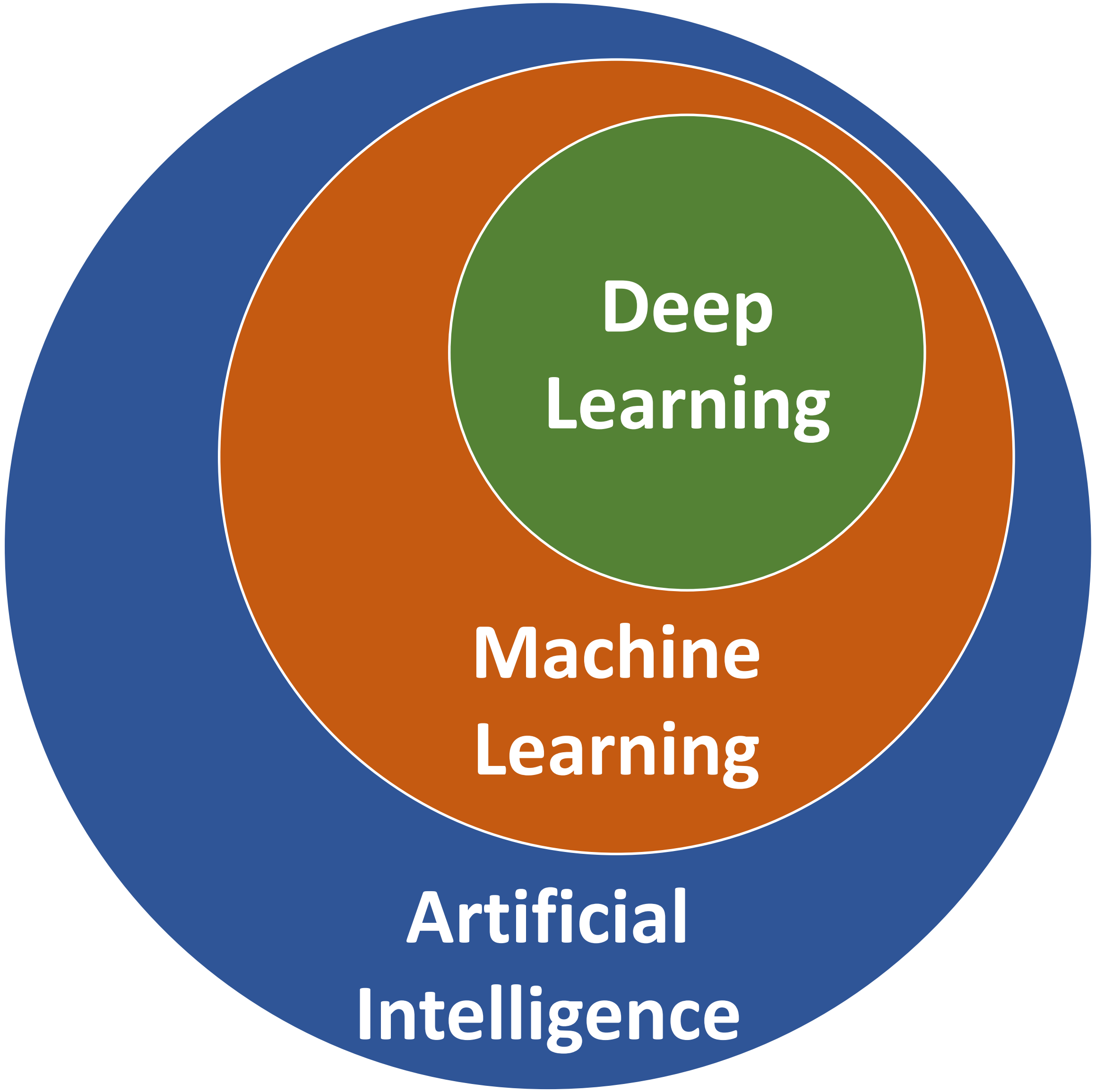
Hands-on Introduction to Deep Learning

Artificial Neural Networks

INSTITUT DU DÉVELOPPEMENT ET DES RESSOURCES EN INFORMATIQUE SCIENTIFIQUE



Algorithms

Health

Finance

NLP

IOT

Media

Business

Autonomous Driving

Virtual agents

Transport

Travelling

Computer Vision

Audio

Application fields

Application fields

IOT

- Personal assistant
- Smart connected object

Healthcare

- Medical Imaging
- Drug research

Finance

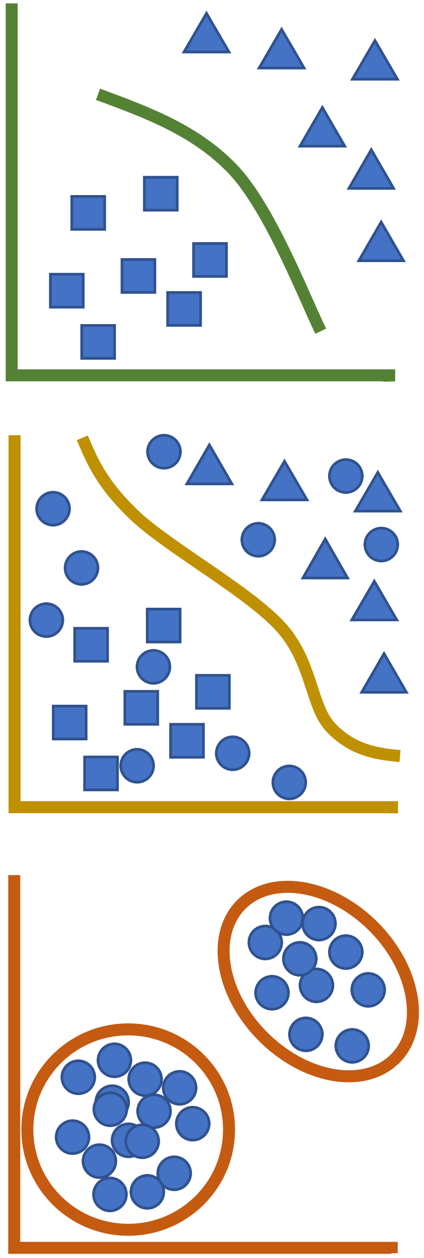
- Fraud detection
- Prediction of financial flows

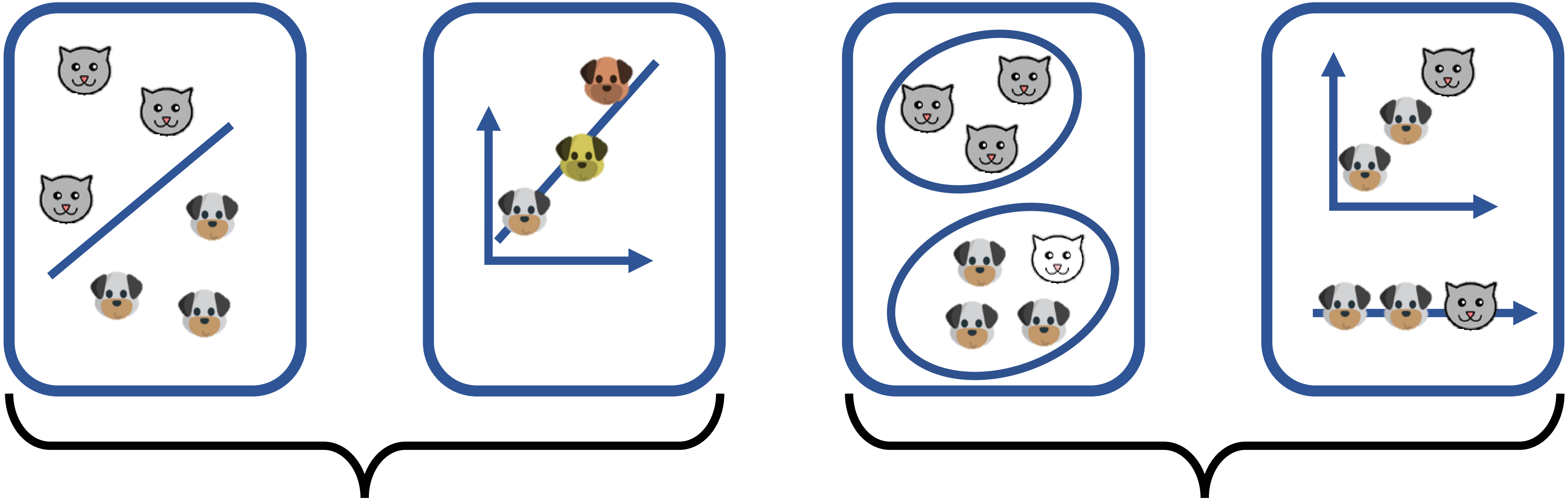
Autonomous driving

- Route optimization
- Autonomous driving

Type of problems

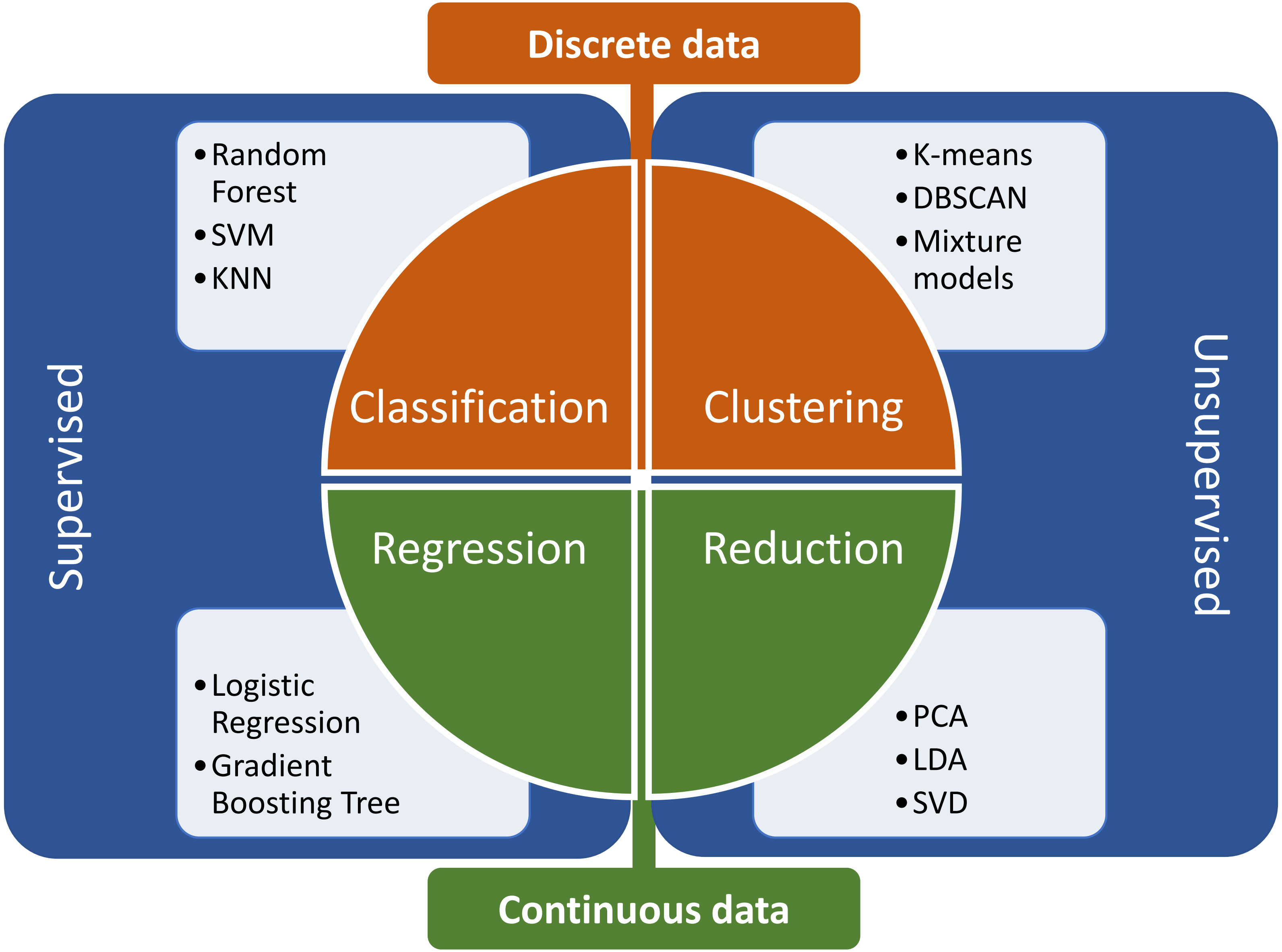
Données	Apprentissage
Labelled	Supervised
Partially labelled	Semi-supervised
Unlabelled	Unsupervised



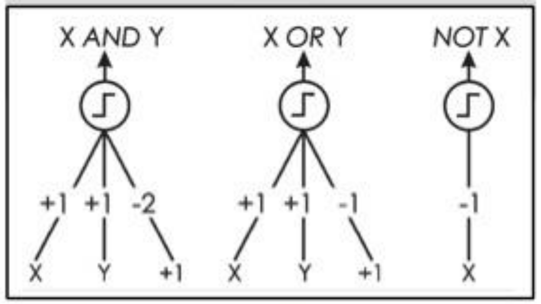


Supervised Learning **Unsupervised Learning**

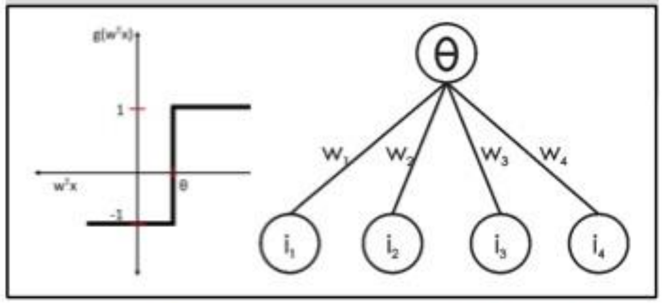
Algorithms


Discrete data
Random Forest
SVM
KNN
K-means
DBSCAN
Mixture models
Supervised
Classification
Clustering
Unsupervised
Regression
Reduction
Logistic Regression
Gradient Boosting Tree
PCA
LDA
SVD
Continuous data

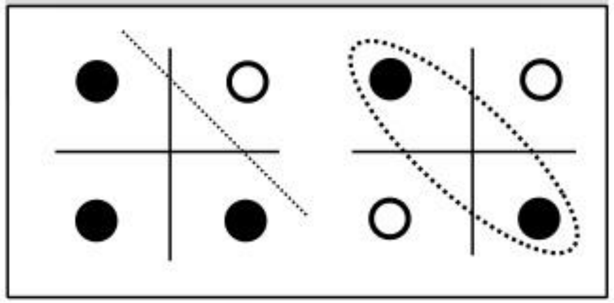
Algorithms



- Adjustable Weights
- Weights are not Learned



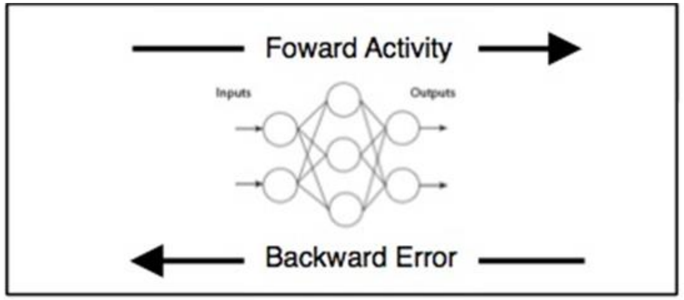
- Learnable Weights and Threshold



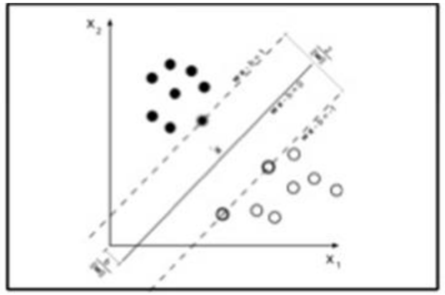
- XOR Problem

40's **Neuron** → **60's** **Networks** → **70-80's** **Winter** → **Concepts** → **Datasets + Hardware** → **10's** **Records**

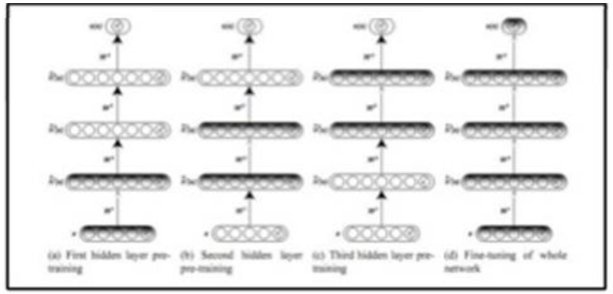
History of Deep Learning



- Solution to nonlinearly separable problems
- Big computation, local optima and overfitting



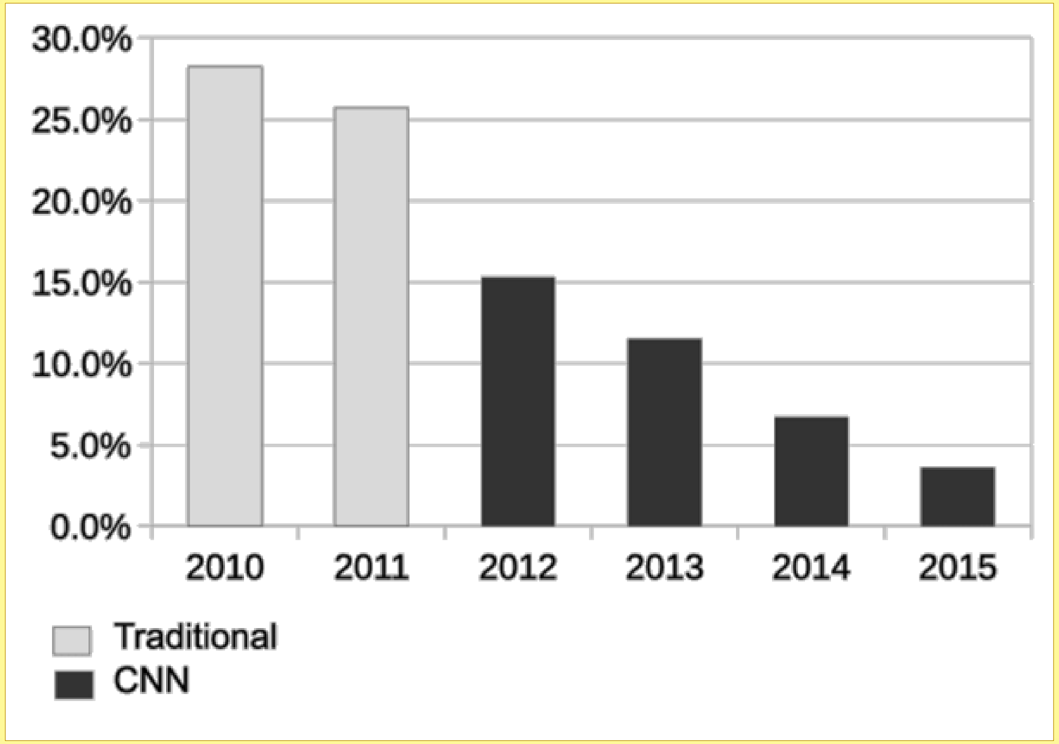
- Limitations of learning prior knowledge
- Kernel function: Human Intervention

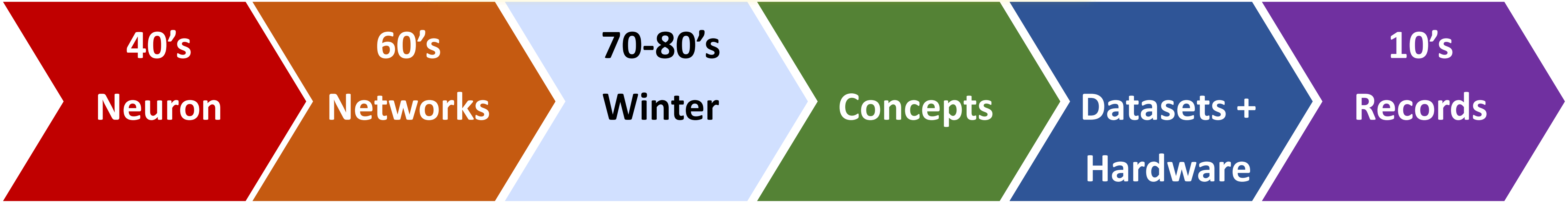


- Hierarchical feature Learning

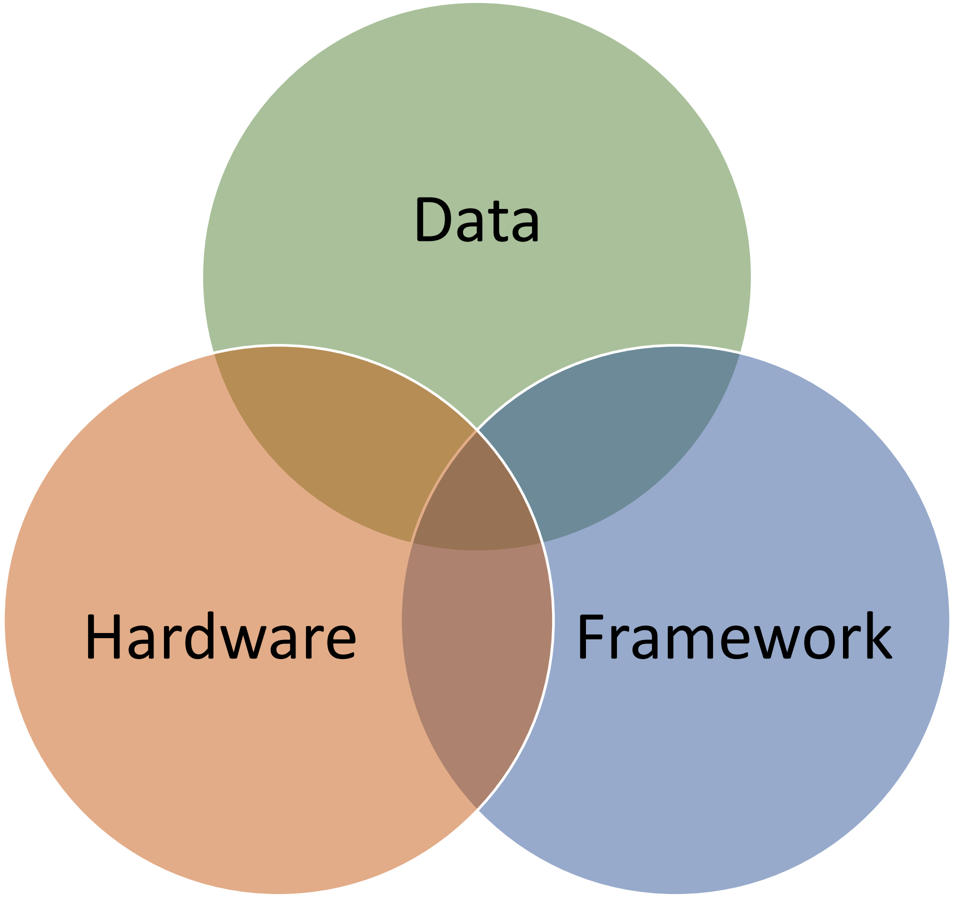
40's	60's	70-80's			10's
Neuron	Networks	Winter	Concepts	Datasets + Hardware	Records

History of Deep Learning

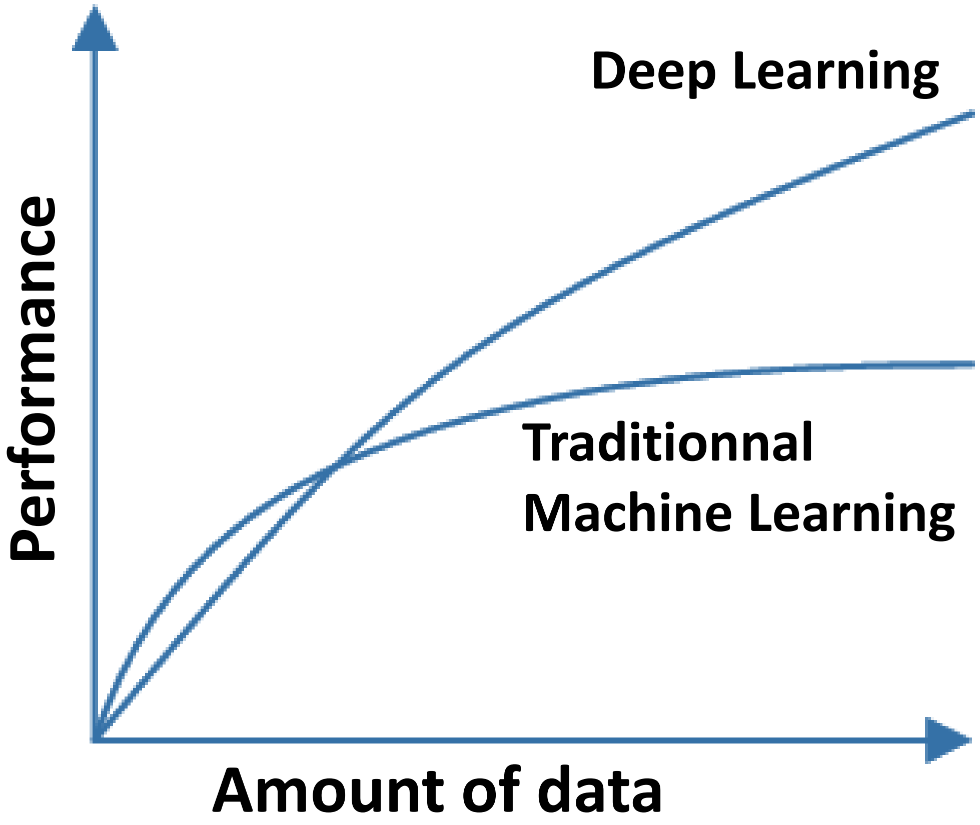




History of Deep Learning



3 decisive factors



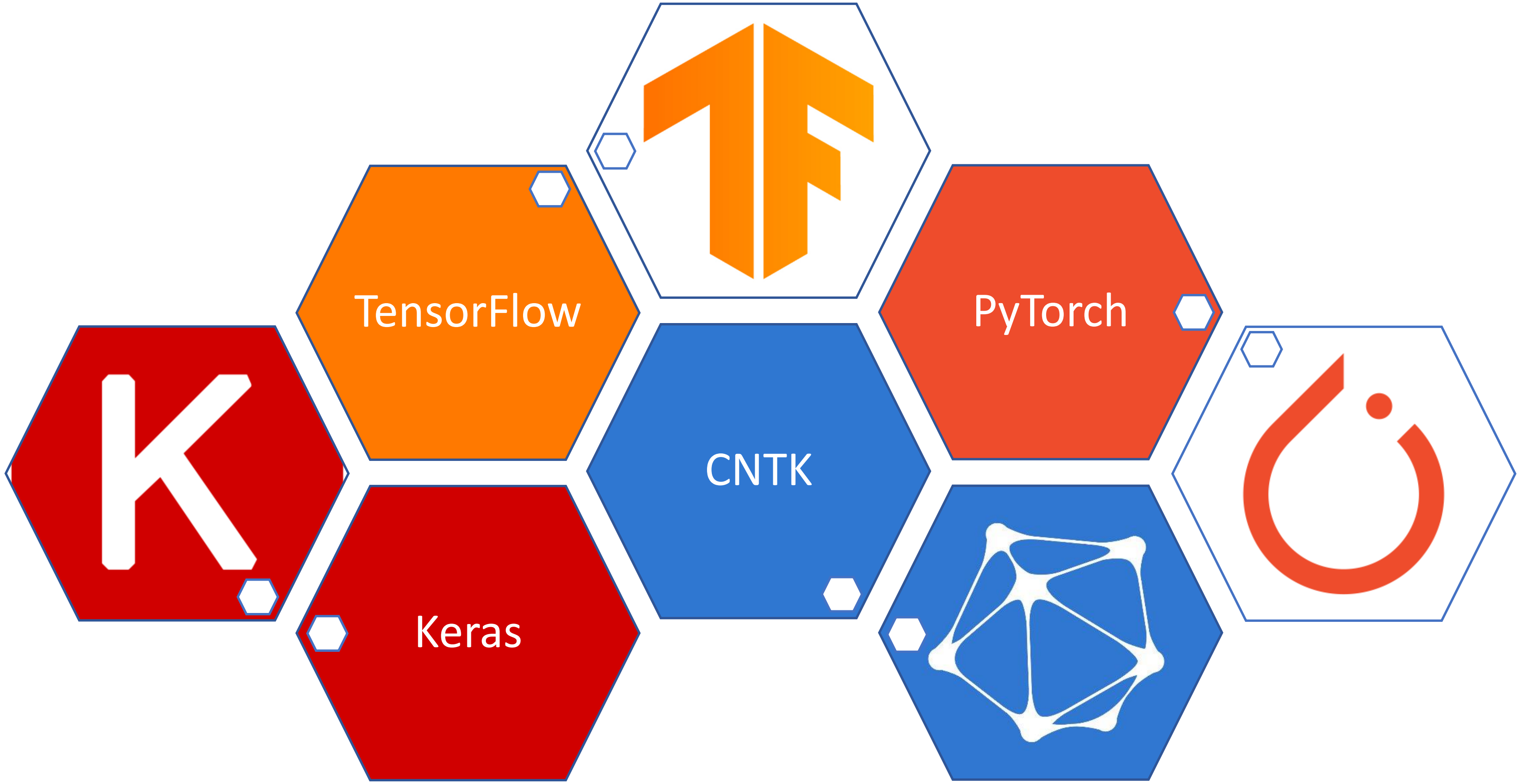
Amount

Open source

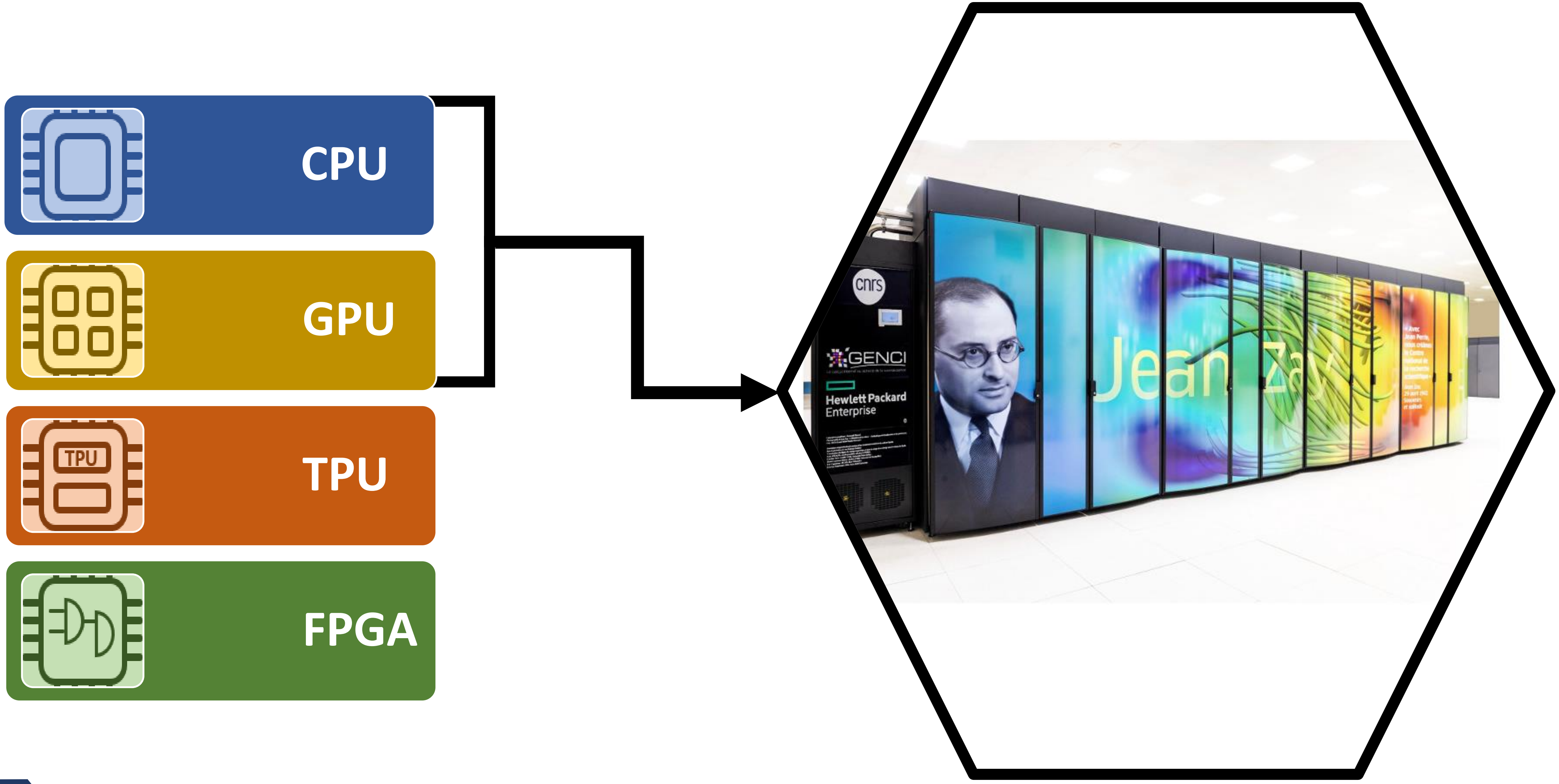
Collaboration

Competitions

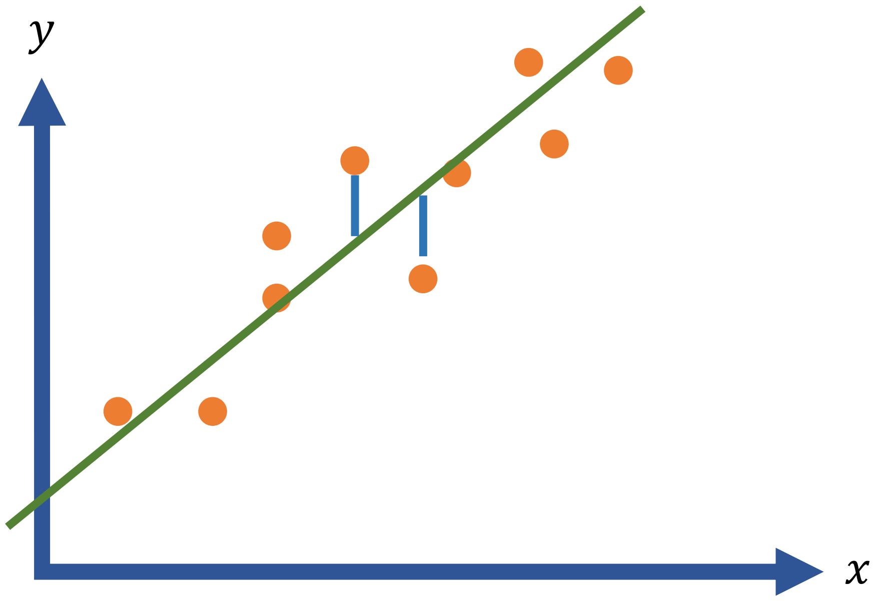
Data



Frameworks and libraries


CPU
GPU
TPU
FPGA
cnrs
GENCI
Hewlett Packard
Enterprise
Jean Zay

Hardware



$$Y = X \cdot \Theta + N \qquad \hat{Y} = X \cdot \widehat{\Theta}$$

With :

$\Theta = (a, b)$

N , noise

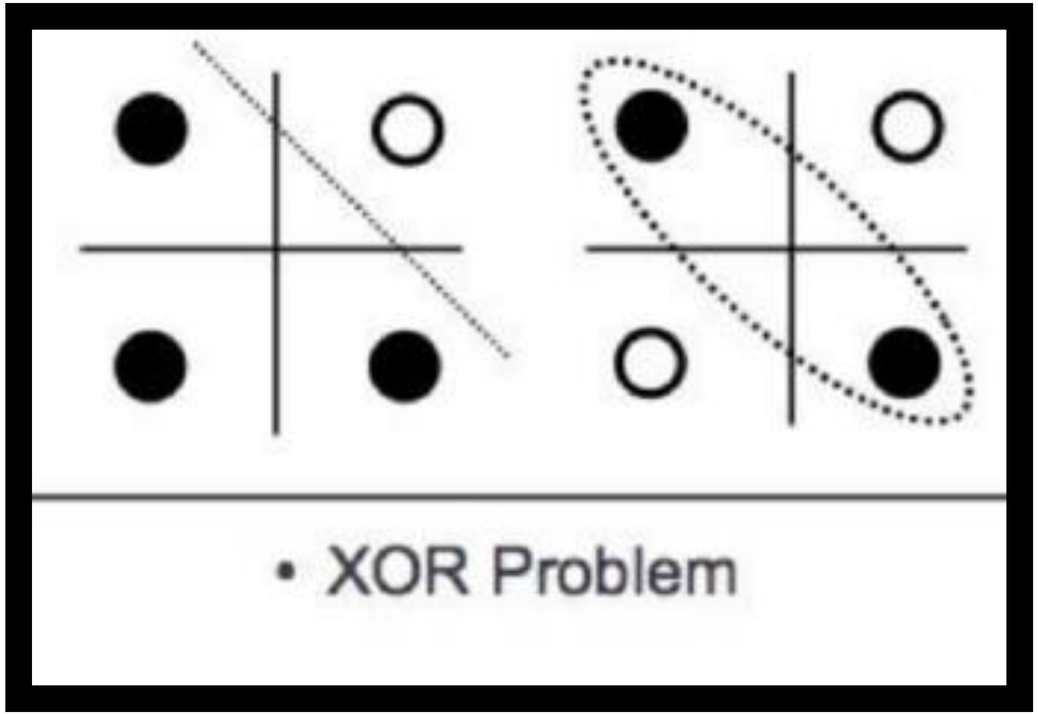
$$\boxed{\min_{\boldsymbol{\theta}} \; J(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2}$$

Cost function

Convex problem : Direct solution

- $\widehat{\Theta} = (X^T \cdot X)^{-1} \cdot X^T \cdot Y$

Linear regression

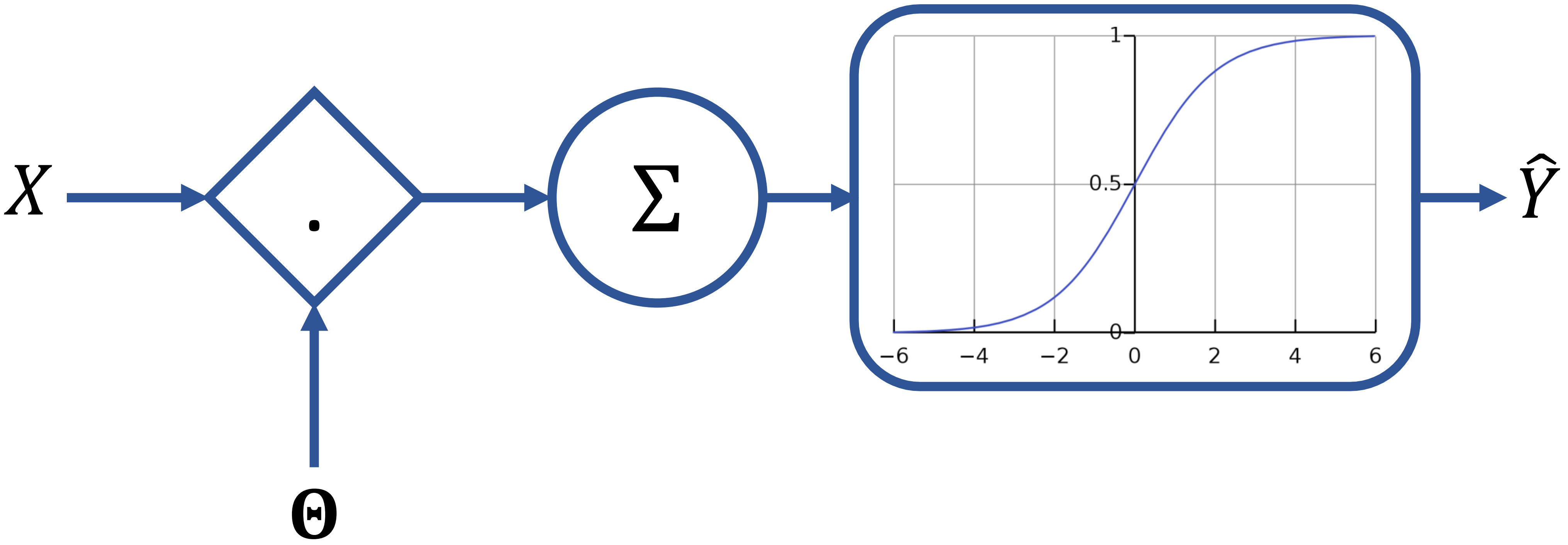


$$f = W\,x$$

$$\widehat{W} = W_2 \cdot W$$
$$\hat{f} = \widehat{W}\,x$$

Non-linear problem?

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

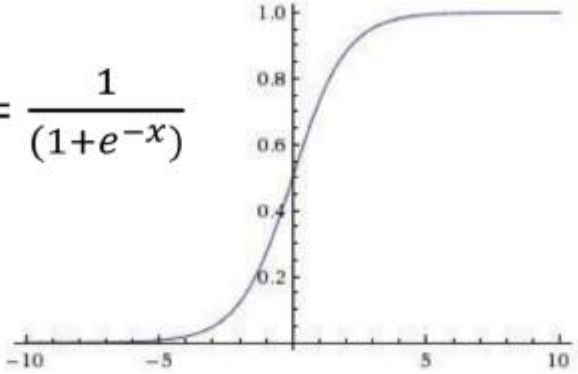


$$\mathcal{L}(\hat{y}_i, y_i) = y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)$$

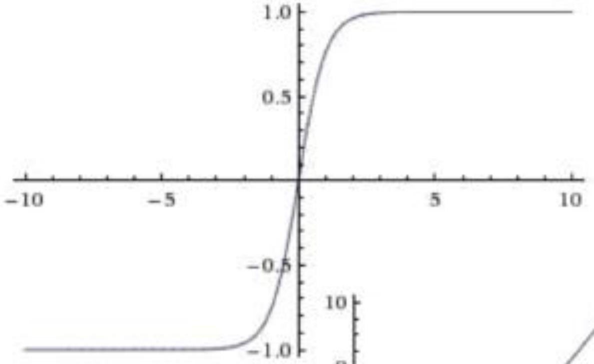
Cross-entropy loss

Logistic regression

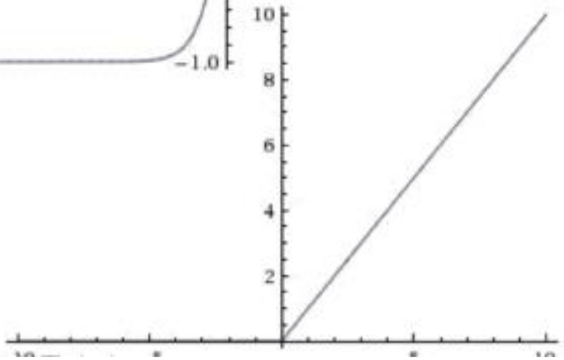
Sigmoid: $\sigma(x) = \frac{1}{(1+e^{-x})}$



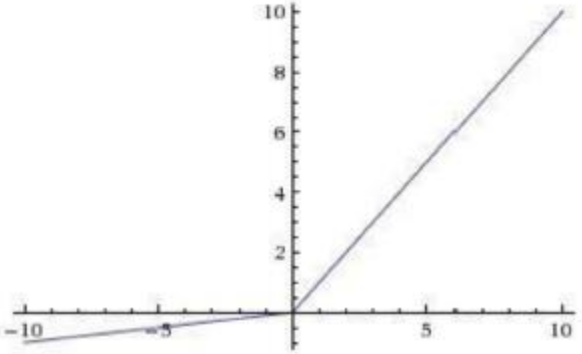
tanh: $\tanh(x)$



ReLU: $\max(0, x)$



Leaky ReLU: $\max(0.1x, x)$

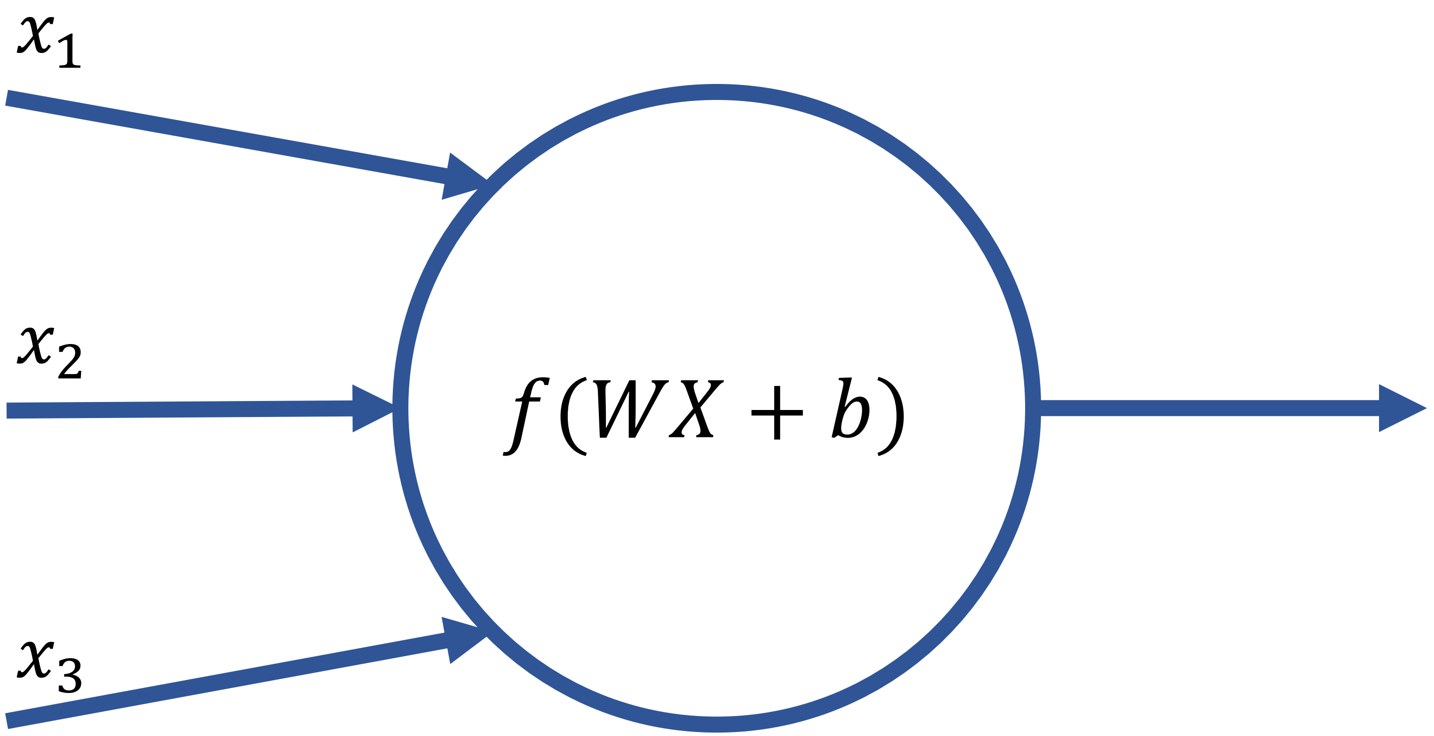


Parametric ReLU: $\max(\alpha x, x)$

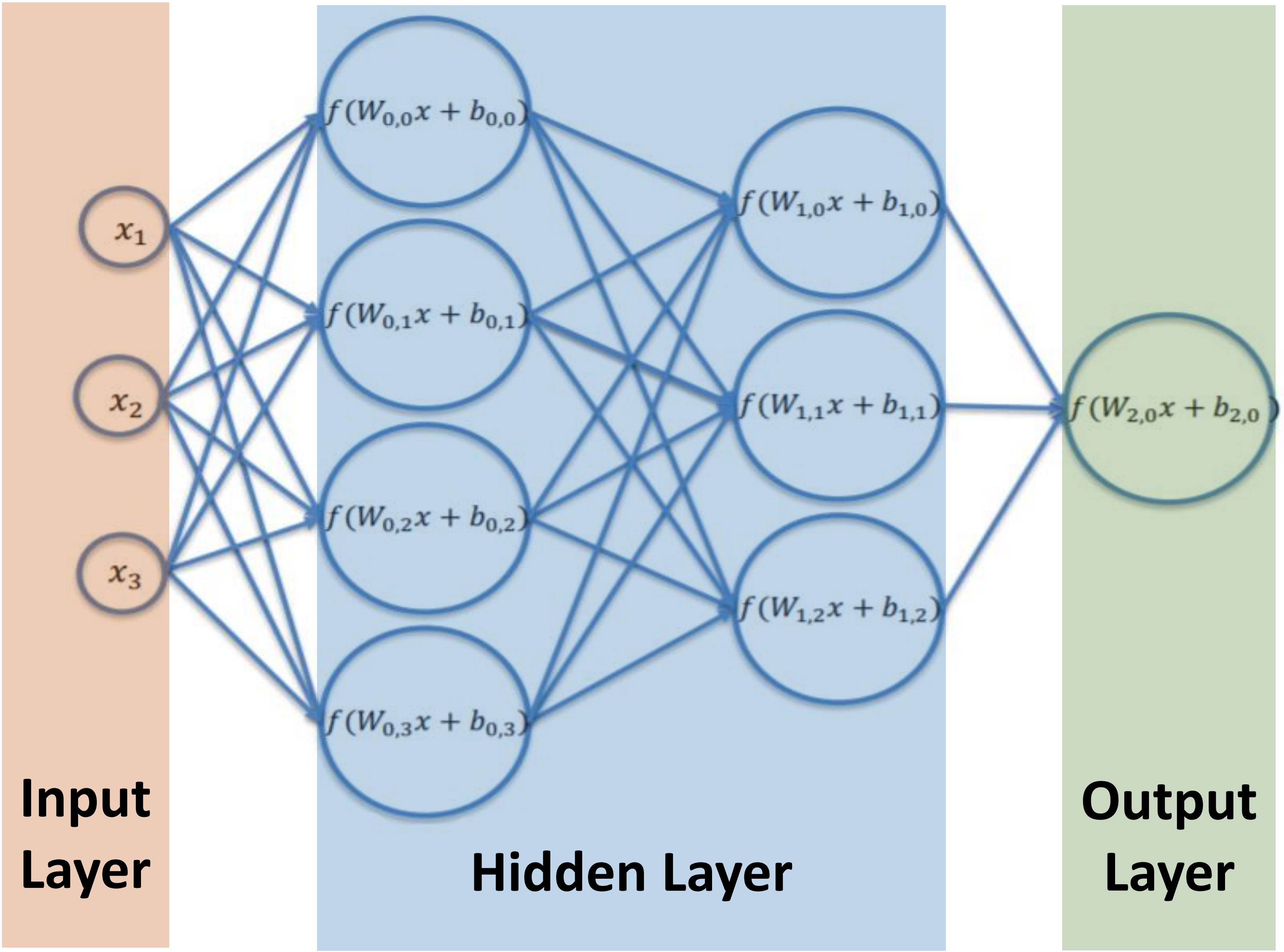
Maxout $\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU $f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \le 0 \end{cases}$

Activation functions



1 neuron



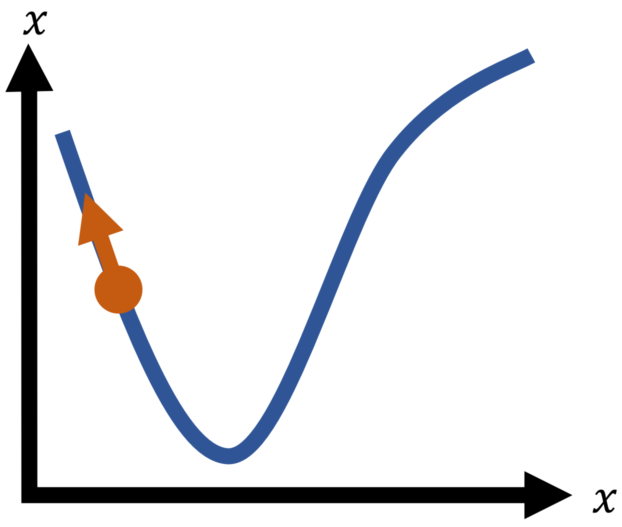
Neural networks

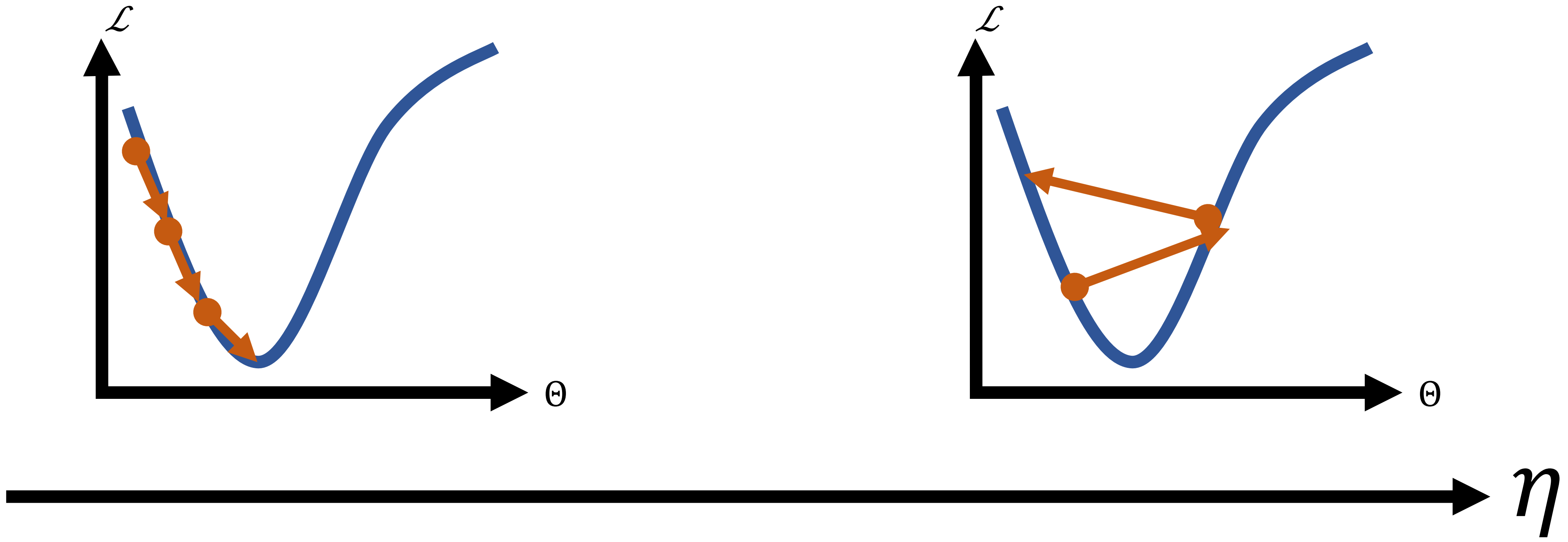
- **Linear systems**
  - LU, QR, Cholesky, Jacobi, Gauss-Seidel, CG, PCG, …
- **Non-linear systems**
  - First order : Gradient Descent, SGD
  - Second order : Newton, Gauss-Newton, LM, (L)BFGS
- **Autres**
  - Genetic algorithms, Metropolis-Hastings, …
  - Complex and constrained solver : ADMM, Primal-Dual, …

Optimization

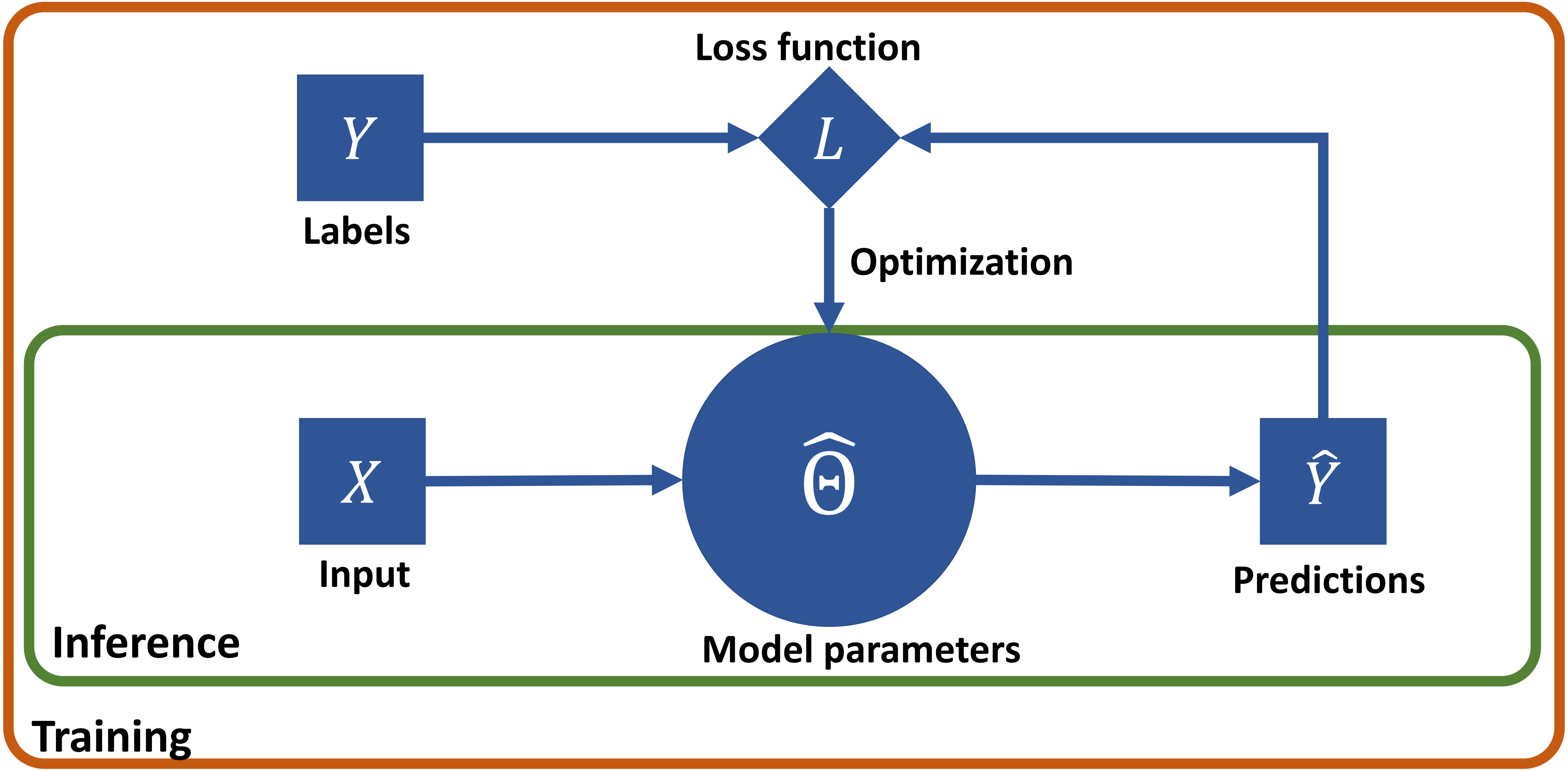
Iterative solution

- Gradient descent
- $\widehat{\Theta}_{t+1} = \widehat{\Theta}_t - \eta \, \nabla_{\boldsymbol{\Theta}} \mathcal{L}(\widehat{y}_i, y_i)$
- η Learning rate





Gradient descent



Training a model

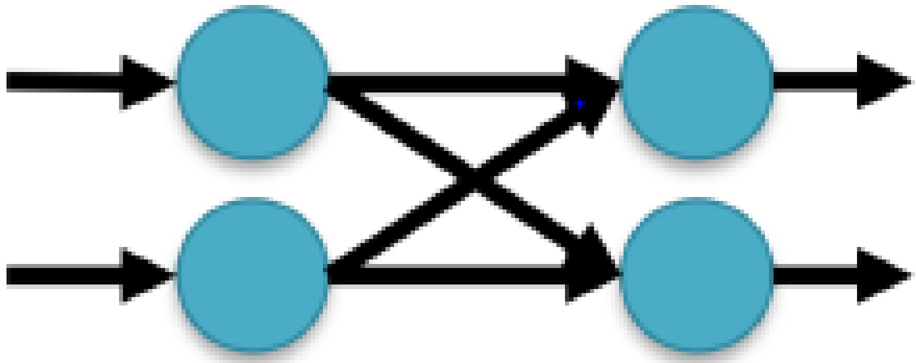
- **Regression loss**
  - Average absolute deviation : $L(y, \hat{y}, \theta) = \frac{1}{n}\sum_i^n |y_i - \widehat{y_i}|$
  - Least squares method : $L(y, \hat{y}, \theta) = \frac{1}{n}\sum_i^n (y_i - \widehat{y_i})^2$
- **Classification loss**
  - Cross-Entropy : $\mathrm{E}(y, \hat{y}, \theta) = -\frac{1}{n}\sum_i^n \sum_j^m y_{ij} \log \widehat{y_{ij}}$

Loss function

$$\widehat{\Theta}_{t+1} = \widehat{\Theta}_t - \eta \nabla_\Theta\left[\mathcal{L}(\widehat{y}_i, y_i) + \lambda R(\widehat{\Theta}_t)\right]$$

L1 : LASSO	L2 : Ridge
$\lvert\Theta\rvert$	Θ^2

Regularization

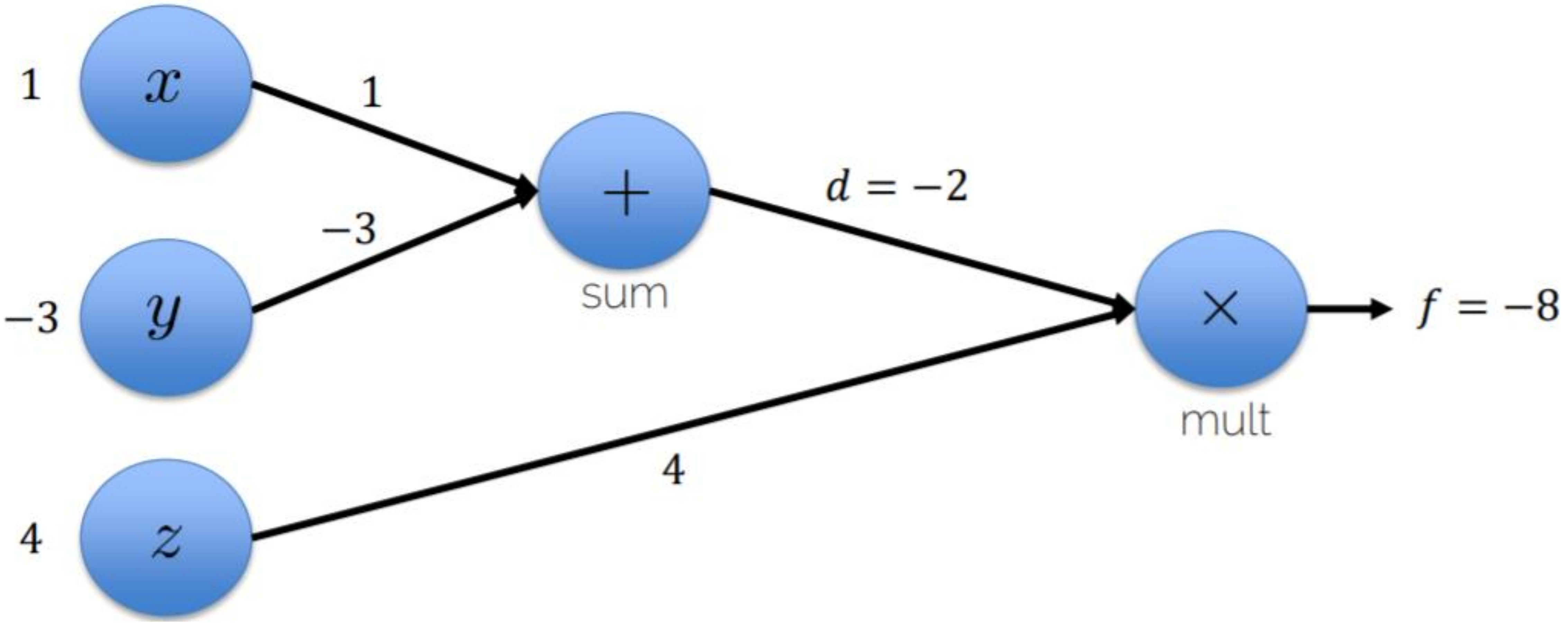


$$\boxed{\frac{\partial f}{\partial x} = \frac{\partial f}{\partial d} \cdot \frac{\partial d}{\partial x}}$$

$$\frac{\partial L}{\partial w_{i,k}} = \frac{\partial L}{\partial \hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial w_{i,k}}$$

Calculation graph and chain rule

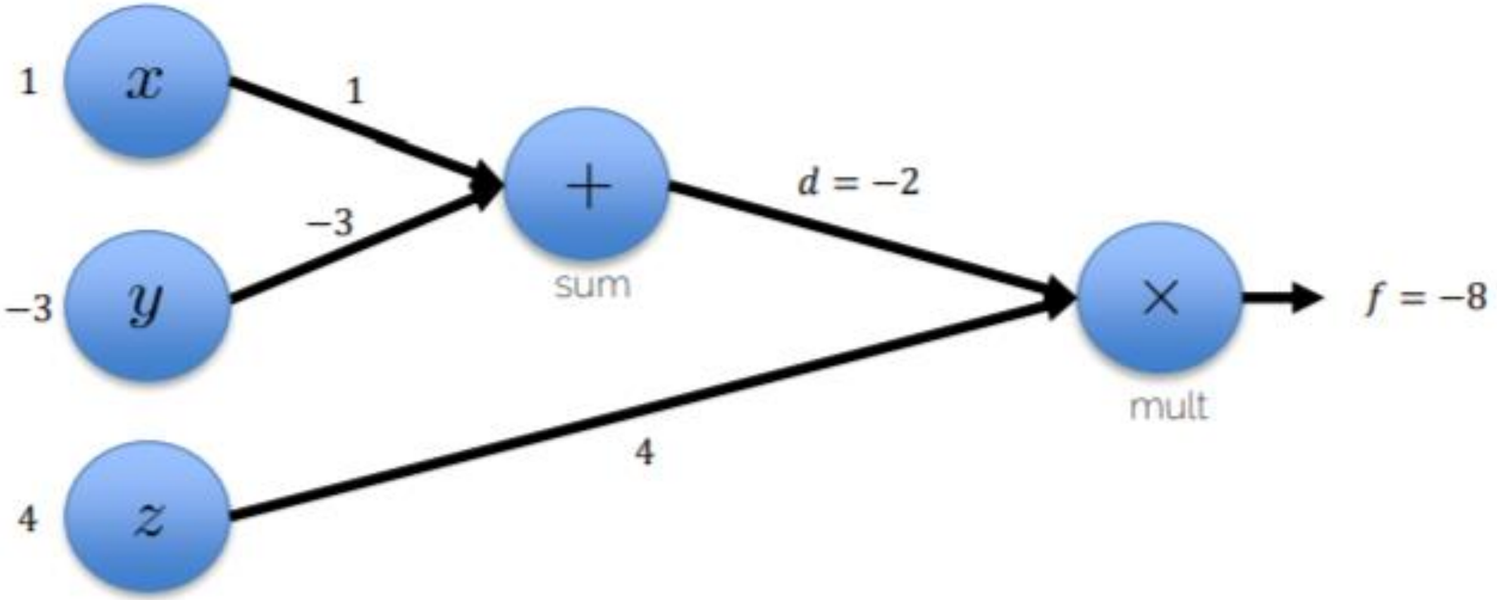
- $f(x, y, z) = (x + y) \cdot z$ Initialization $x = 1, y = -3, z = 4$



Forward pass

$$f(x,y,z) = (x+y)\cdot z$$

with $x = 1, y = -3, z = 4$



$$d = x + y \qquad \frac{\partial d}{\partial x} = 1,\ \frac{\partial d}{\partial y} = 1$$

$$f = d \cdot z \qquad \frac{\partial f}{\partial d} = z,\ \frac{\partial f}{\partial z} = d$$

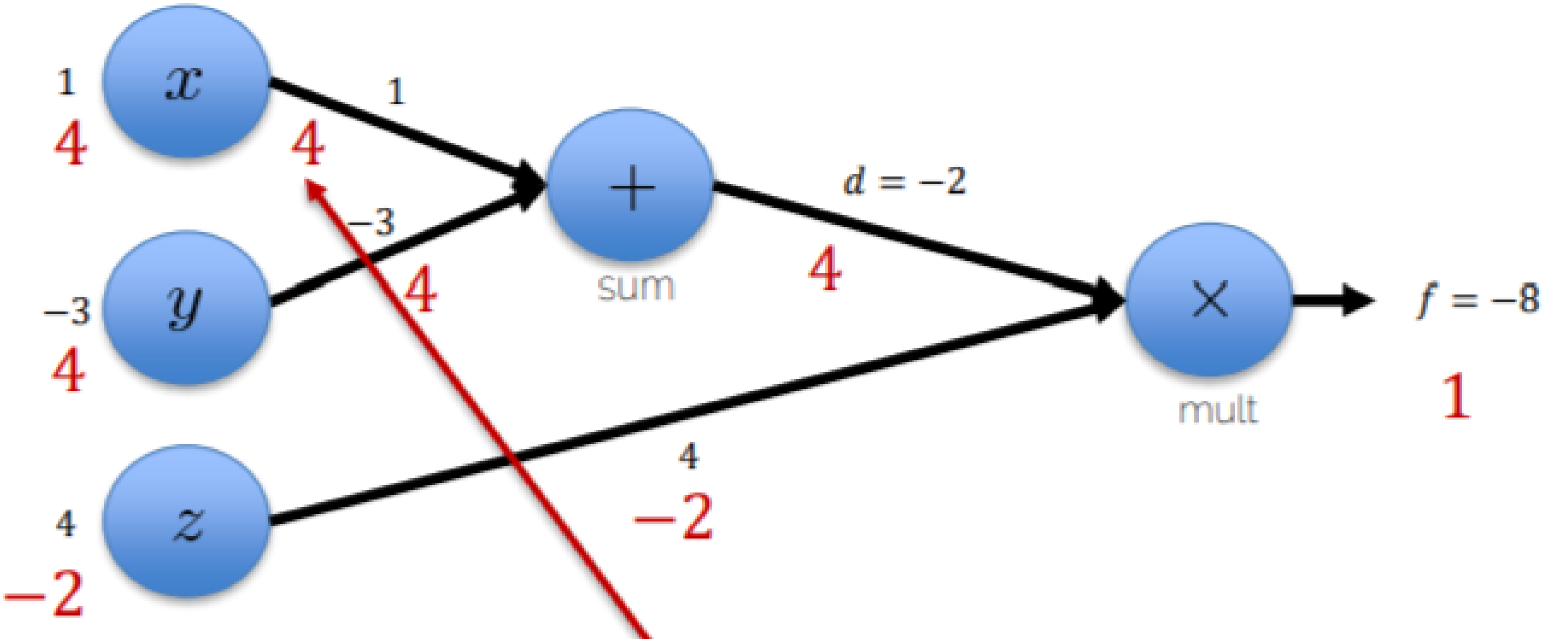
What is $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$?

Backward pass

Backward pass

$$f(x,y,z) = (x+y)\cdot z$$

with $x = 1, y = -3, z = 4$



$d = x + y \qquad \boxed{\dfrac{\partial d}{\partial x} = 1} \quad \dfrac{\partial d}{\partial y} = 1$

$f = d \cdot z \qquad \dfrac{\partial f}{\partial d} = z, \quad \dfrac{\partial f}{\partial z} = d$

Chain Rule:

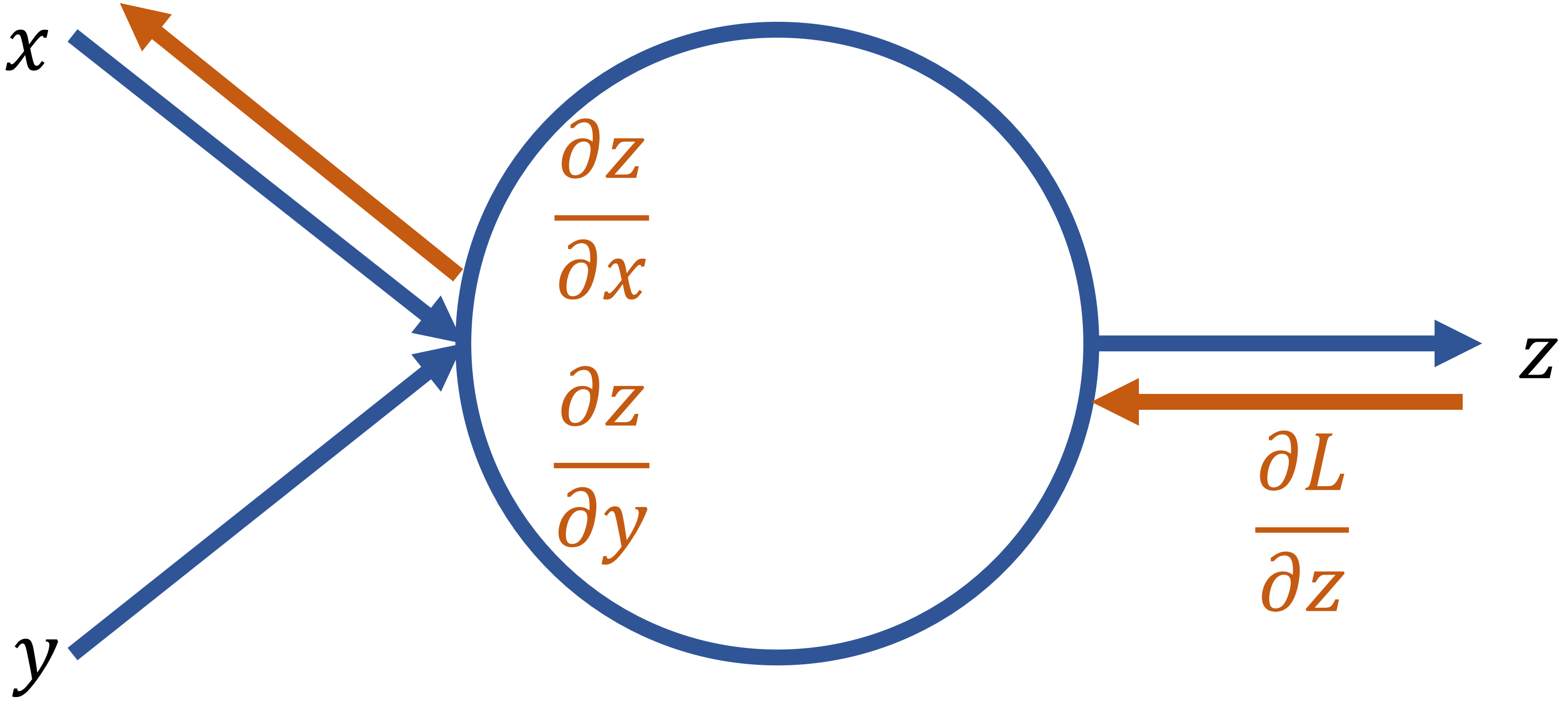
$$\boxed{\frac{\partial f}{\partial x} = \frac{\partial f}{\partial d} \cdot \frac{\partial d}{\partial x}}$$

$$\boxed{\frac{\partial f}{\partial x}}$$

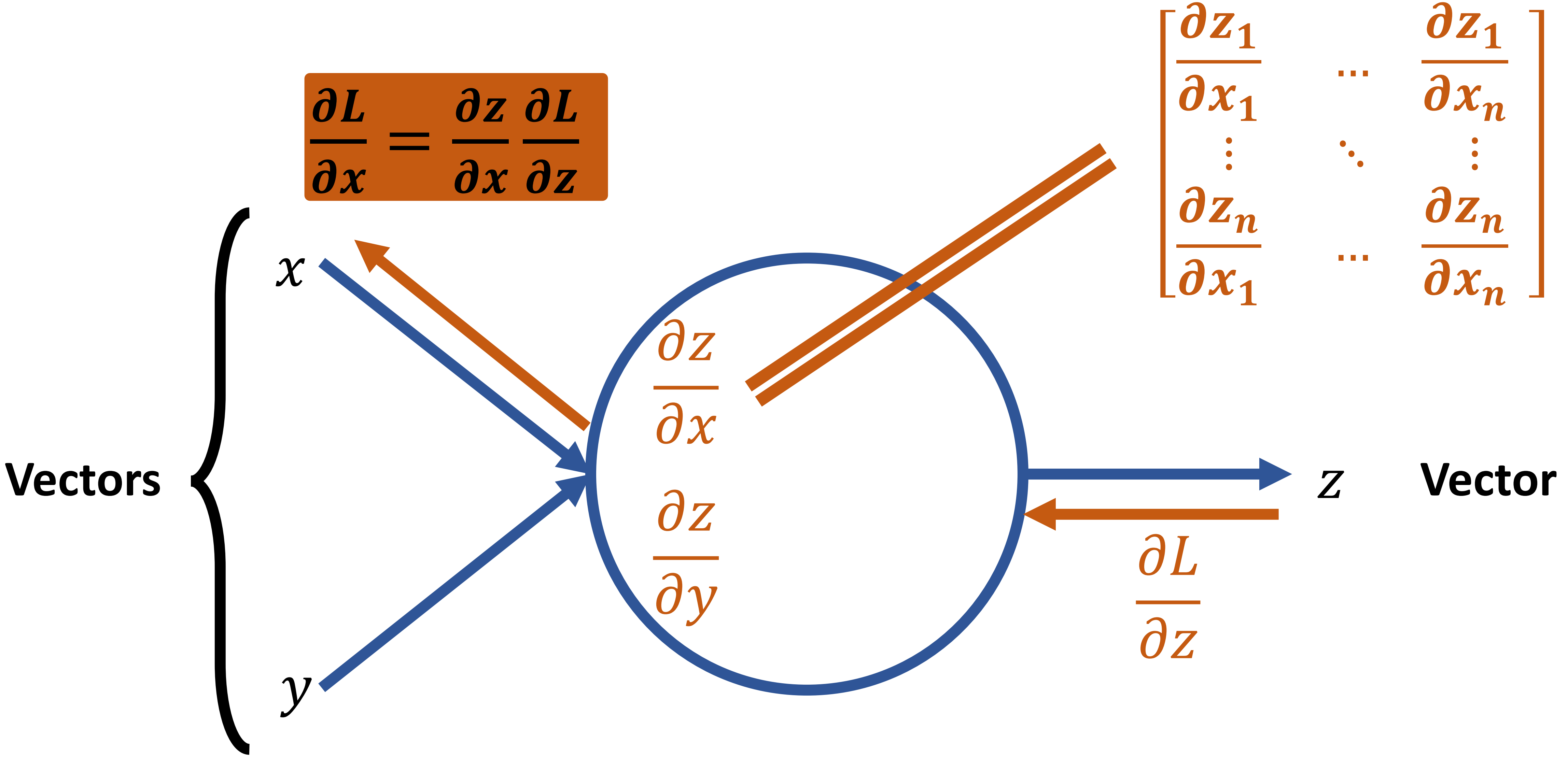
$$\rightarrow \frac{\partial f}{\partial x} = 4 \cdot 1 = 4$$

What is $\dfrac{\partial f}{\partial x}, \dfrac{\partial f}{\partial y}, \dfrac{\partial f}{\partial z}$?

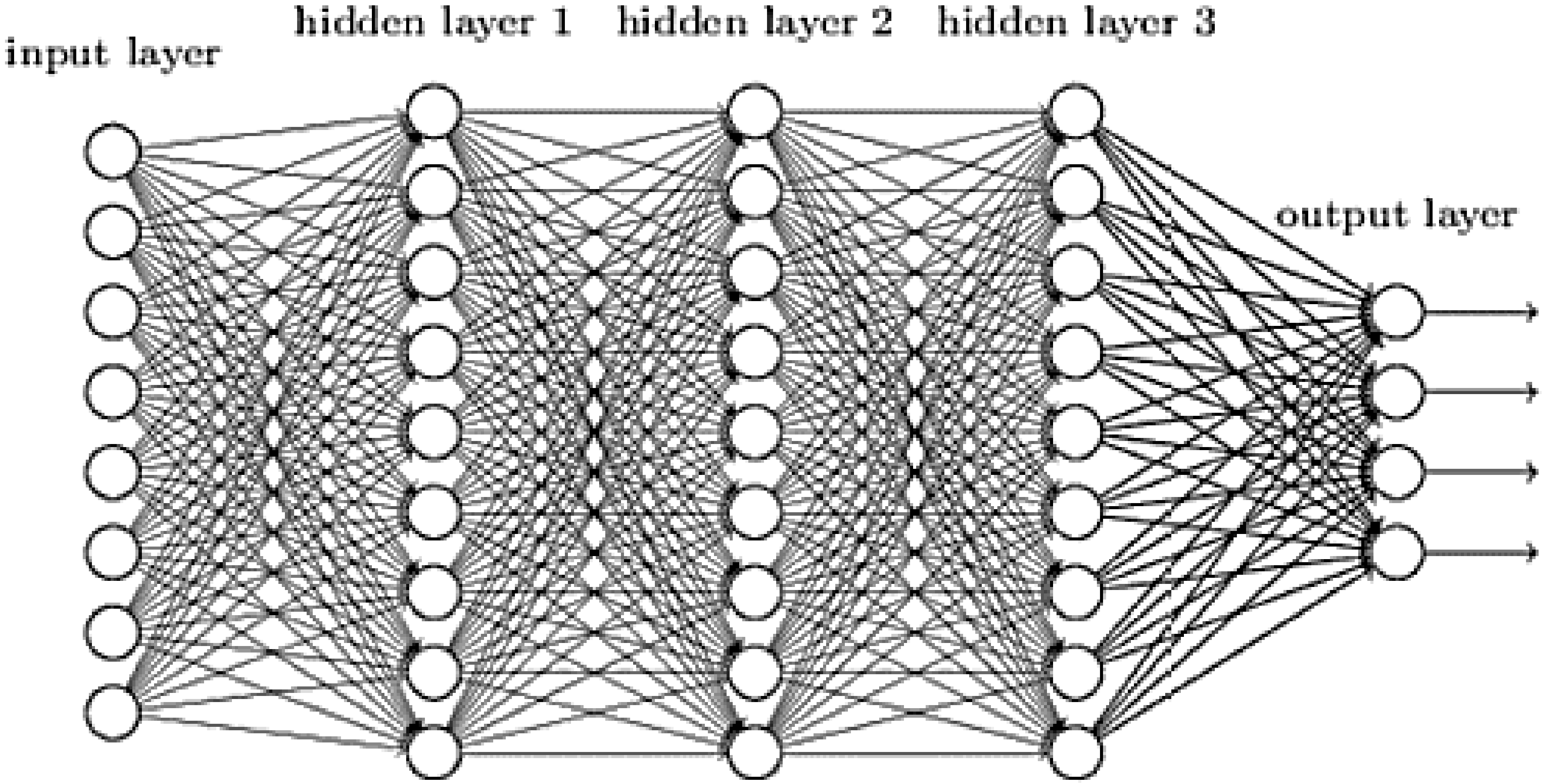
$$\frac{\partial L}{\partial x} = \frac{\partial z}{\partial x}\frac{\partial L}{\partial z}$$



Gradient Flow



Gradient Flow



Network size

- Chollet, Francois. *Deep learning with Python*. Simon and Schuster, 2021.
- *CS230 Deep Learning*. cs230.stanford.edu. Accessed 14 Mar. 2022.
- *I2DL*. niessner.github.io/I2DL. Accessed 14 Mar. 2022.
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

References